

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

BAKALÁŘSKÁ PRÁCE

2014

Ondřej Lazarczyk

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
K a t e d r a i n f o r m a t i k y

**Detekce obsazených a volných míst na parkovišti
pomocí příznakového rozpoznání**

Parking Lot Occupation Detection
using Feature-Based Detectors

Zadání bakalářské práce

Student: **Ondřej Lazarczyk**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Detekce obsazených a volných míst na parkovišti pomocí příznakového rozpoznání**
Parking Lot Occupation Detection using Feature-Based Detectors

Zásady pro vypracování:

Detekce obsazenosti parkovacích míst může mít v dnešní době celou řadu využití. Jedním z takových využití může být informace o obsazenosti daného parkoviště, která se pošle řidiči do automobilu. Ten se poté rozhodne, zda si toto parkoviště vybere. Tato detekce je obecně zajišťována pomocí různých senzorů. Cílem této práce je vytvořit detektor obsazenosti parkovacích míst za pomoci kamery snímající parkovací plochu. Tento detektor bude pracovat na základě příznaků založených na informaci o jasu, gradientu a histogramu.

1. V teoretické části se zaměřte na detekční metody, které využívají gradientní informace pro popis objektů, také se seznamte se základními gradientními operátory.
2. Seznamte se ze základními klasifikátory (zejména se zaměřte na SVM).
3. Po získání teoretických znalostí implementujte algoritmus pro detekci obsazených a volných parkovacích míst za pomoci příznaků, které ponesou informaci o jasu, gradientu a histogramu a vektor těchto příznaků bude sloužit jako vstup pro klasifikátor.
4. Otestujte úspěšnost takového přístupu.
5. Své závěry řádně zdokumentujte v textu práce.

Seznam doporučené odborné literatury:

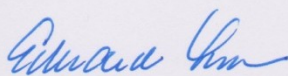
Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 886-893 vol. 1 (june 2005)

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Radovan Fusek**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

Datum odevzdání: 2.5.2014

Podpis: Ondřej Lazarec

Rád bych vyjádřil poděkování vedoucímu Ing. Radovanu Fuskovi za odbornou pomoc a cenné rady při vypracování této bakalářské práce.

Abstrakt

V dnešní době může mít detekce obsazenosti parkovacích míst různá využití. Například může být informace o obsazenosti zobrazena řidiči v jeho automobilu nebo může být zobrazena u parkoviště. Tato informace může být poté využita řidičem automobilu k rozhodnutí se, kde bude parkovat. K detekování obsazenosti parkovacích míst je využíváno mnoho různých senzorů a jejich kombinací. Využitím kamerového systému je umožněno monitorovat parkoviště bez nutnosti dalších senzorů. V této práci je cílem vytvoření detektoru pro určení zda je parkovací místo volné nebo obsazené za pomoci kamery, která snímá parkoviště. Detektor je založen na informaci o jasu, na základě které je provedena detekce hran a z výsledků je pak vypočítán histogram. Ke klasifikaci je použit klasifikátor Support Vector Machine (SVM). Práce obsahuje experimenty s parametry algoritmu použitého pro detekci obsazenosti parkovacích míst.

Klíčová slova

extrakce příznaků; detekce objektů; SVM; HOG; SIFT

Abstract

In recent years, detection of the parking places can be used in many positive ways. For example, the detection results may be displayed inside the cars or at the parking lot. Based on the detection results, the drivers can easily find free places. In general, the parking lot places can be detected using many sensors. With the use of camera, the parking lot can be monitored very efficiently, therefore, the camera based parking lot detector is presented in this paper. This detector is based on the gradient information which is encoded using the histograms. The histograms are composed to the feature vector that is used as an input for the SVM classifier. The detection results and parameters of the method are presented in the work.

Key Words

feature extraction; object recognition; SVM; HOG; SIFT

Obsah

Úvod.....	7
1 Detekce hran	8
1.1 Gradientní metody pro detekci hran v obraze	8
1.2 Metoda detekce hran hledáním průchodu nulou.....	11
2 Histogram	14
3 Algoritmy pro detekci	15
3.1 Scale-Invariant Feature Transform (SIFT).....	15
3.2 Histogramy orientovaných gradientů.....	16
4 Klasifikátory	17
4.1 Support Vector Machine (SVM)	17
4.1.1 Lineární SVM.....	17
4.1.2 Kernel.....	19
4.1.3 Nelineární SVM.....	20
4.2 Další klasifikátory	20
5 Popis použitého algoritmu	22
5.1 Proces trénování	26
6 Experimenty.....	27
6.1 Použité operátory.....	27
6.2 Kombinace detekcí hran a vyhlazení	28
6.2.1 Vyhlazení před použitím Sobelova operátoru	29
6.2.2 Vyhlazení před použitím Laplaceova operátoru	29
6.3 Velikost bloků	30
6.4 Parametr C u lineárního SVM	33
6.5 Chybně pozitivní a negativní.....	34
6.6 Čas detekce parkoviště.....	34
Závěr	35
Použitá literatura.....	36
Seznam obrázků.....	38
Seznam tabulek.....	39
Seznam příloh.....	40

Úvod

V současnosti je mnoho detekčních systémů založených na různých typech senzorů nahrazováno kamerovými systémy. Detekce za pomoci kamerového systému je využívána pro detekci pohybu, osob a objektů. Cílem této práce je vytvoření detektoru, který bude podávat informaci o obsazenosti parkoviště. Tato informace o obsazenosti parkoviště může být využita řidičem automobilu, aby se dozvěděl, zda je na parkovišti volné místo k parkování přímo z auta. Například z mobilní aplikace, sdělující zda je na parkovišti místo nebo může být obsazenost zobrazena na informační tabuli před samotným parkovištěm. K detekci je využíváno snímků parkoviště pořízených z kamerového systému umístěného na střeše budovy v areálu Vysoké školy báňské – Technické univerzity v Ostravě. Detektor využívá informaci o jasu, která je použita k výpočtu detekce hran a ta je následně převedena do podoby histogramu. Extrakce příznaků pro detektor je provedena následujícím algoritmem. RGB obraz parkovacího místa je převeden do odstínů šedi. Nad obrazem v odstínech šedi je provedena detekce hran a výsledek vzniklý detekcí je rozdělen na bloky. V každém bloku je vypočítán histogram. Zkombinováním těchto histogramů do vektoru je vytvořen vektor příznaků, který je poté využit v klasifikátoru Support Vector Machine (SVM) k určení zda je místo obsazené nebo prázdné. V práci byly provedeny a zdokumentovány experimenty s různými způsoby detekce hran, velikostmi bloků a nastaveními klasifikátoru.

1 Detekce hran

Algoritmus použitý k detekci obsazenosti parkovacích míst využívá detekci hran. Výsledek detekce je následně využit k popisu tvaru a vzhledu objektů. V experimentech byly použity dva různé operátory pro detekci hran. Každý je založen na jiné metodě. V následujících kapitolách jsou tyto metody popsány.

1.1 Gradientní metody pro detekci hran v obraze

Gradientní metody využívají toho, že v místě, kde se v obraze nachází hrana, je vysoká hodnota absolutní hodnoty první derivace průběhu jasu přes hranu. Hodnota derivace určuje velikost hrany. Derivace $\frac{\partial f}{\partial x}$ a $\frac{\partial f}{\partial y}$ popisují změnu jasu ve směru osy x a osy y a jsou pravděpodobně nejjednodušším způsobem určení velikosti hrany. Operátory používané pro určení velikosti hrany se jmenují hranové operátory. Tyto operátory jsou schopné nalézt hrany, které jsou rovnoběžné se souřadnými osami. Průběh jasu napříč hranou při zjišťování hran, které mají obecný směr, musí být v kolmém směru na hranu, která je detekována. Tento kolmý směr je vyjádřen jako vektor $n = (\cos \theta, \sin \theta)$. Pomocí derivace ve směru je možné zjistit, jestli existuje v daném bodě hrana určeného směru. Vzorec pro derivaci ve směru je

$$\frac{\partial f}{\partial \xi} = \text{grad}(f) \times n = \frac{\partial f}{\partial x} \cos \theta + \frac{\partial f}{\partial y} \sin \theta. \quad (1.1)$$

Kde ξ je souřadnice měřená ve směru. Absolutní hodnota z $\frac{\partial f}{\partial \xi}$ v daném bodě je velikostí hrany. Protože předem není známo, jaký směr má hledaná hrana, tak je nejdříve zjištěn směr gradientu obrazové funkce, který odpovídá směru, ve kterém je největší změna jasu. Kolmý směr ke směru gradientu určuje směr hrany. Pro zjednodušení zápisu je zavedeno označení $f_x(x, y) = \frac{\partial f(x, y)}{\partial x}$ a $f_y(x, y) = \frac{\partial f(x, y)}{\partial y}$. Pokud je velikost hrany označena jako $e(x, y)$, a ta odpovídá velikosti gradientu v bodě (x, y) , potom $e(x, y)$ je vypočteno jako

$$e(x, y) = \sqrt{f_x^2(x, y) + f_y^2(x, y)}. \quad (1.2)$$

V bodě (x, y) je směr gradientu označen jako $\varphi(x, y)$ a směr hrany jako $\Psi(x, y)$. Pro jejich výpočet jsou použity tyto vzorce

$$\varphi(x, y) = \arctan \left[\frac{f_x(x, y)}{f_y(x, y)} \right], \quad \Psi(x, y) = \varphi(x, y) + \frac{\pi}{2}. \quad (1.3)$$

V praxi se nejčastěji pracuje s obrazovou funkcí, která je diskrétní. Protože zatím byla obrazová funkce spojitá, je potřeba nahradit derivace diferencemi. Pokud je tento převod proveden pro vztahy $f_x(x, y)$ a $f_y(x, y)$ je pro diskrétní případ dosaženo vztahů

$$f_x(x, y) = f(x + 1, y) - f(x, y), \quad (1.4)$$

$$f_y(x, y) = f(x, y + 1) - f(x, y). \quad (1.5)$$

Velikost hrany je v praktickém výpočtu porovnávána s hodnotou prahu ve všech bodech obrazu.

Operátory pro výpočet velikosti hrany sloužící k detekci hran:

- Robertsův operátor

Jeden z nejstarších operátorů používaných pro detekci hran. Výpočet velikosti hrany v obraze je dán diferencí ve dvou směrech a to v navzájem kolmých a zároveň diagonálních. Výpočet v bodě (x, y) je proveden vztahem

$$e(x, y) = \sqrt{(f(x, y) - f(x + 1, y + 1))^2 + (f(x + 1, y) - f(x, y + 1))^2}. \quad (1.6)$$

- Operátor Prewittové

Operátor využívá obrazové funkce ve všech pixelech, které sousedí s bodem (x, y) , k výpočtu velikosti hrany. Stejně jako předchozí operátor počítá difference ve dvou směrech a to průměrem ve směru osy x a y . Označením těchto sousedících pixelů jako A, B, C, D, F, G, H, I jsou získány tyto předpisy

$$f_x(x, y) = \frac{1}{3}[(C - A) + (F - D) + (I - G)], \quad (1.7)$$

$$f_y(x, y) = \frac{1}{3}[(A - G) + (B - H) + (C - I)]. \quad (1.8)$$

Postup využívající výpočet diferencí průměrem snižuje vzniklý šum při detekci.

Tabulka 1.1: Označení sousedících pixelů pro operátor Prewittové

A	B	C
D	$f(x, y)$	F
G	H	I

- Sobelův operátor

Derivace v jednotlivých směrech jsou při použití Sobelova operátoru zprůměrovány váženým průměrem, jinak je podobný operátoru od Prewittové. Derivace jsou počítány ve dvou směrech a to směru osy x a osy y.[1]

Vztahy pro výpočet jsou následující

$$f_x(x, y) = \frac{1}{4}[(C - A) + 2(F - D) + (I - G)], \quad (1.9)$$

$$f_y(x, y) = \frac{1}{4}[(A - G) + 2(B - H) + (C - I)]. \quad (1.10)$$

Také je možné výpočet provést konvolucí obrazové funkce $f(x, y)$ s konvoluční maskou. Před provedením výpočtu je originální obraz převeden do odstínu šedi. Obrázek 1.1 je ukázkou tohoto převodu.

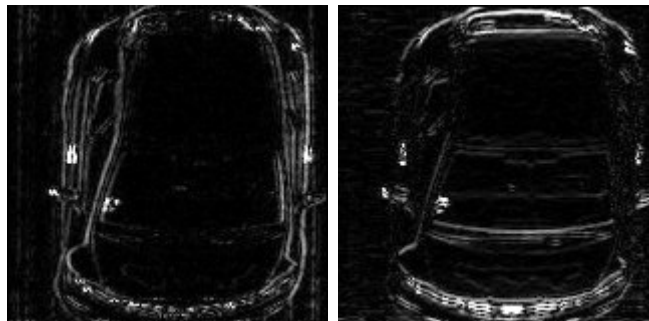


Obrázek 1.1: Převedení originálního obrazu (vlevo) do odstínů šedi (vpravo)

Konvoluce pro horizontální směr $f_x(x, y) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * f(x, y)$ a pro vertikální směr

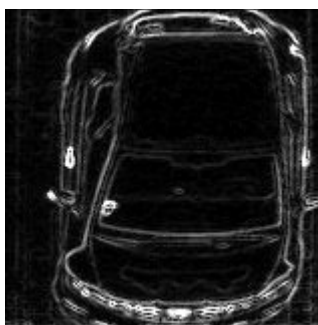
$$f_y(x, y) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * f(x, y).$$

Výsledky po provedení výpočtů konvolucí v obou směrech nad obrázkem 1.1 vpravo jsou zobrazeny na obrázku 1.2.



Obrázek 1.2: Výsledek konvolučních masek pro horizontální (vlevo) a vertikální (vpravo) směr

Výsledná velikost gradientu je počítána pomocí vzorce pro výpočet velikosti gradientu.[2]



Obrázek 1.3: Výsledek po vypočtení velikosti gradientu

1.2 Metoda detekce hran hledáním průchodu nulou

Tato metoda je využívána Laplaceovým operátorem, který k detekování hrany používá druhé derivace průběhu jasů přes hranu. V místě hrany tato druhá derivace prochází nulou a ve směru průchodu hranou nabývá dvou extrémů opačného znaménka. Tak jako při detekci hran pomocí gradientních metod tak i při zjišťování hran hledáním průchodů nulou v druhé derivaci většinou předem není znám směr hrany. Proto je druhá derivace počítána ve dvou směrech a to obvykle ve směrech souřadných os.

Označením druhých derivací jako $f_{xx}(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2}$ a $f_{yy}(x, y) = \frac{\partial^2 f(x, y)}{\partial y^2}$, je následně operátor vyjádřen vztahem

$$\nabla^2 f(x, y) = f_{xx}(x, y) + f_{yy}(x, y). \quad (1.11)$$

Pro diskrétní obrazovou funkci, po nahrazení derivací diferencemi, jsou získány vzorce

$$f_{xx}(x, y) = f(x - 1, y) - 2f(x, y) + f(x + 1, y), \quad (1.12)$$

$$f_{yy}(x, y) = f(x, y - 1) - 2f(x, y) + f(x, y + 1). \quad (1.13)$$

S využitím vztahů 1.11, 1.12, 1.13 předešlých vzorců vznikne výsledný předpis Laplaceova operátoru

$$\nabla^2 f(x, y) = f(x - 1, y) + f(x + 1, y) + f(x, y - 1) + f(x, y + 1) - 4f(x, y). \quad (1.14)$$

Podobně jako u Sobelova operátoru je možné provést výpočet pomocí konvoluce obrazové funkce $f(x, y)$ s konvoluční maskou, která je pouze jedna sloužící pro výpočet jak v směru osy x tak ve směru osy y .

Tabulka 1.2: Konvoluční maska Laplaceova operátora

0	1	0
1	-4	1
0	1	0

Problém nastává v rozích, krajních řádcích a sloupcích obrazu kde uvedená konvoluční maska nemůže být použita. Pro tyto speciální případy je použita konvoluční maska zobrazena v tabulce 1.3 sloužící pro výpočet hodnot dolního řádku v obraze.

Tabulka 1.3: Konvoluční maska pro výpočet hodnot v dolním řádku obrazu

0	1	0
1	-3	1
0	0	0

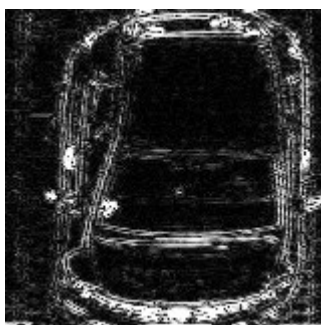
Výpočet hodnoty v levém dolním rohu obrazu je proveden pomocí konvoluční masky v tabulce 1.4.

Tabulka 1.4: Konvoluční maska pro výpočet v levém horním rohu obrazu

0	1	0
0	-2	1
0	0	0

Masky pro ostatní okrajové řádky, sloupce a rohy jsou získány rotací masek v tabulkách 1.3, 1.4. Pokud jsou v obraze nějaké hrany, tak se ve výsledcích operátoru budou vyskytovat extrémy opačného znaménka. Pokud mezi dvěma takovými extrémy bude výsledek z Laplaceova operátoru měnit znaménko, tak tato místa budeme považovat za hranu v obraze.

Při použití této metody pro detekci hran je potřeba řešit problémy s citlivostí na šum, tato citlivost vede k detekování nepřehlédnutelného počtu neexistujících hran i při malém zašumění v obraze. Tyto problémy je možné do jisté míry snížit pomocí filtrace šumu, která je aplikována na obraz před tím, než jsou detekovány hrany. Často používanou metodou bývá konvoluce obrazové funkce s Gaussiánem a až poté je provedena aplikace Laplaceova operátoru. Na obrázku 1.4 je možné vidět výsledek detekce hran za pomoci Laplaceova operátoru i s výše popsáním problémem s šumem.[1]



Obrázek 1.4: Výsledek detekce hran Laplaceovým operátorem

2 Histogram

Vzhledem k použití výpočtu histogramu k detekci obsazenosti parkovacích míst a jeho přítomnosti v následující kapitole jsou zde uvedeny základní informace o histogramu.

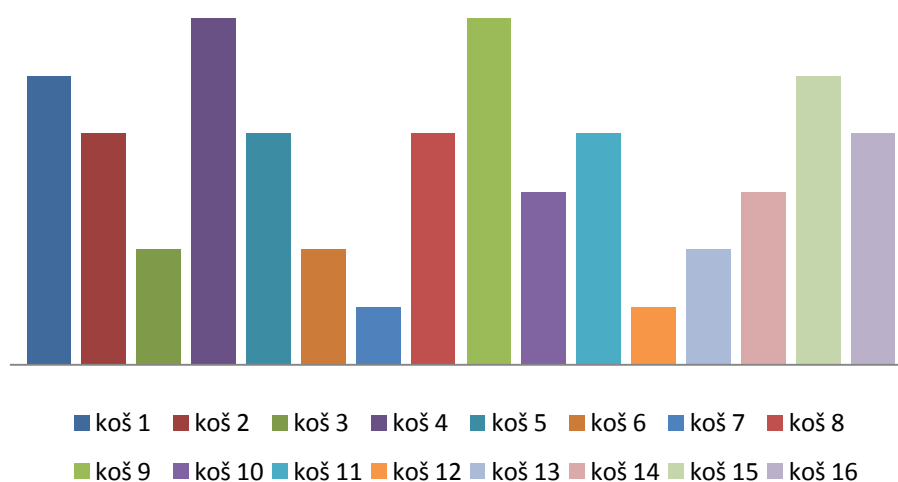
Histogramy slouží k získání statistického znázornění rozložení dat. Jsou to počítačové data a tyto počty jsou následně organizovány do předdefinovaných košů. Pro naplnění košů počty je možné použít vlastnosti vypočítané z dat. Tyto vlastnosti mohou být například velikost a směr gradientu, jas nebo jakákoliv jiná vlastnost, která je užitečná k popisu obrazu.[3]

K názornému příkladu je použita matice, která obsahuje informace o obraze. Informací je intenzita v rozsahu 0-255. Pokud je potřeba nějak zorganizovat výskyt dat, je možné rozdělit rozsah intenzity do úseků nazývaných koše. Například takto

$$[0,255] = [0,15] \cup [16,31] \cup \dots \cup [240,255], \quad (2.1)$$

$$\text{rozsah} = \text{bin}_1 \cup \text{bin}_2 \cup \dots \cup \text{bin}_{16}. \quad (2.2)$$

Poté je pro každý koš počítán počet pixelů, které spadají do jeho rozsahu. Pokud je takto rozdělen a spočítán počet pixelů v jednotlivých koších podle uvedeného příkladu a výsledek je zobrazen pomocí grafu, vypočtený histogram by mohl vypadat jako na obrázku 2.1. Osa x v obrázku znázorňuje jednotlivé koše a osa y počet pixelů, které spadají do rozsahu jednotlivých košů.[4]



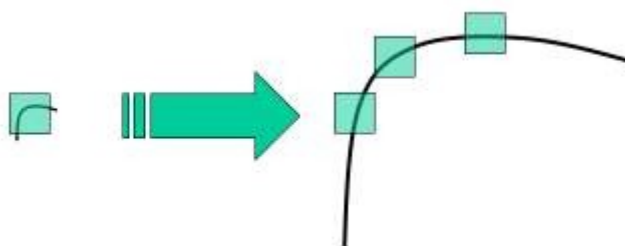
Obrázek 2.1: Grafické znázornění vypočteného histogramu

3 Algoritmy pro detekci

Algoritmus, který byl použit pro detekci obsazenosti parkovacích míst na parkovišti, byl inspirován algoritmem Histogramy orientovaných gradientů (HOG), který použili Navneet Dalal a Bill Triggs ve své práci [5] pro detekci lidí. Tento algoritmus je modifikací algoritmu Scale-Invariant Feature Transform (SIFT) sloužícímu k detekci objektů. Z tohoto důvodu je jako první popsán algoritmus SIFT a následně HOG.

3.1 Scale-Invariant Feature Transform (SIFT)

Rohové detektory jsou schopné detekovat stejné rohy i v otočeném obraze, ale pokud se změní velikost obrazu tak se může stát, že roh v malém obraze uvnitř malého okna se po přiblížení ve stejném okně zarovná, jak můžete vidět na obrázku 3.1.



Obrázek 3.1: Ukázka problému nastávajícího při detekci rohů

Řešení takového problému bylo nalezeno spolu s algoritmem Scale-Invariant Feature Transform (SIFT). Algoritmus se skládá z čtyř hlavních kroků, které jsou dále popsány:

- Detekce extrému prostoru velikostí

Jelikož není možné detekovat důležité body různých velikostí pomocí stejných oken, je potřeba pro větší body větší okna. Pro vyhledání důležitých bodů různých velikostí se použije rozdíl Gaussiánů (DoG - Difference of Gaussians). Poté následuje zjištění lokálního maxima a minima pomocí porovnání s jeho osmi sousedy a devíti pixely předchozí a následující velikosti. Pokud je pixel lokální minimum nebo maximum stává se z něj potencionální důležitý bod.[6]

- Lokalizace důležitých bodů

Pro jednotlivé potencionální body je použit detailnější model určující jejich přesnější umístění a velikost. Následně jsou vybrány důležité body na základě jejich stability.

- Přiřazení orientace

Všem důležitým bodům je pomocí vypočtení směru gradientu v lokálním obraze přiřazena jedna nebo více orientací. Aby se dosáhlo nezávislosti na měřítku, umístění a orientaci, jsou všechna data transformována relativně vzhledem ke změnám, na kterých je vyžadována nezávislost.

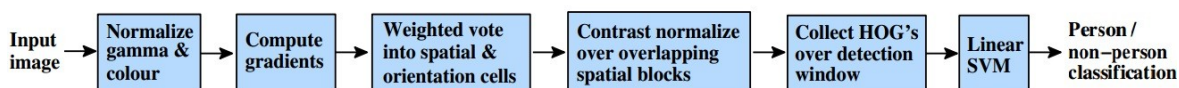
- Deskriptor důležitých bodů

V měřítku důležitého bodu se v jeho okolí vypočítá velikost gradientu a orientace. Pro co nejmenší vliv deformace tvaru objektů a změnám v osvětlení jsou data převedena do reprezentace, která toto umožňuje. Touto reprezentací je deskriptor tvořený normalizovaným vektorem, který obsahuje histogramy orientací a každý histogram má více košů, které odpovídají různým směrům.[7]

3.2 Histogramy orientovaných gradientů

Deskriptory Histogramu orientovaných gradientů (HOG) jsou příznakové deskriptory, kterých se využívá k rozpoznávání objektů v obraze. Jejich základní myšlenkou je, že lze použít rozložení intenzity gradientu nebo orientace hrany k popisu vzhledu objektů a jejich tvaru. Následující sekce popisuje postup pro získání příznaků HOG deskriptorů.

V praxi vytvoření deskriptoru začíná rozdělením obrazu na menší, navzájem na sebe navazující, části obrazu, které se nazývají buňky. Následuje vypočtení histogramu orientovaných gradientů nebo hranové orientace pro pixely náležící buňce. Tyto výpočty se provedou pro každou buňku zvlášť. Zkombinováním výsledků histogramů jednotlivých buněk je získán popis obrazu. Pro zlepšení neměnnosti při změnách osvětlení a stínování se využívá rozdělení obrazu na hustou mřížku navzájem se překrývajících částí, které jsou větší než buňky, nazývané bloky. Bloky se použijí pro vypočítání rozsahu intenzity a tento rozsah potom slouží k normalizaci všech buněk, které náleží do daného bloku. Tyto normalizované bloky deskriptoru se nazývají deskriptory Histogramu orientovaného gradientu (HOG). Příznakový vektor skládající se z jednotlivých HOG deskriptorů se použije jako vstup pro Support Vector Machine (SVM) klasifikátor.[5,8]



Obrázek 3.2: Grafické znázornění HOG algoritmu

4 Klasifikátory

4.1 Support Vector Machine (SVM)

Tento klasifikátor byl použit pro klasifikaci při detekci obsazenosti parkovacích míst. Je algoritmem spadajícím do skupiny jádrových algoritmů (kernel machines), které umožňují zároveň reprezentovat složité nelineární funkce a při tom dovolují využívat výhod, které poskytují efektivní lineární metody.

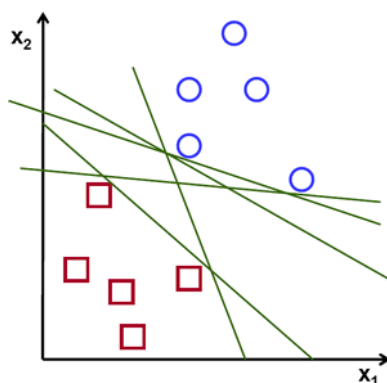
4.1.1 Lineární SVM

Lineárně rozdělitelná data

SVM je diskriminační klasifikátor, který provádí binární klasifikaci pomocí vytvoření rozdělovací nadroviny. Jeho výsledkem je optimální nadrovina, která je dána vztahem

$$f(x) = \beta^T x + \beta_0, \quad (4.1)$$

kde β je váhový vektor a β_0 je práh. V 2D prostoru jsou data rozdělena přímkou a v 3D prostoru je rozdělovačem rovina. Například pro obrázek 4.1 existuje mnoho přímek, pomocí kterých by bylo možné data spadající do dvou rozdílných tříd rozdělit.



Obrázek 4.1: Několik možností pro rozdělení dvou tříd přímkou

Tento optimální rozdělovač je získán na základě úvahy, která říká, že rozdělovače probíhající příliš blízko bodů nebudou správně generalizovat a budou náchylné na šum. Proto je za optimální rozdělovač považován ten, který má největší minimální vzdálenost od trénovacích dat.

Dvojnásobek této vzdálenosti je nazýván pásmo (margin). Z toho vyplývá, že cílem optimálního rozdělovače je maximalizovat toto pásmo. Protože je možné definovat stejnou nadrovinu nekonečným počtem změn velikostí β a β_0 byla vybrána reprezentace základní nadroviny daná předpisem

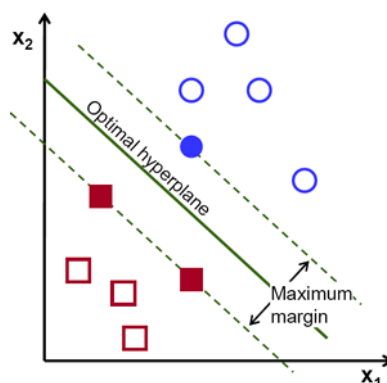
$$|\beta^T x + \beta_0| = 1. \quad (4.2)$$

Podpůrné vektory jsou trénovací data, která se nacházejí nejbližě nadrovině a v předchozím předpisu jsou označená písmenem x . V obrázku 4.2 jsou tyto podpůrné vektory označeny plnými čtverci a plným kruhem. S využitím předchozího předpisu je možné vypočítat vzdálenost podpůrných vektorů od nadroviny jako

$$distance = \frac{|\beta^T x + \beta_0|}{\|\beta\|} = \frac{1}{\|\beta\|}. \quad (4.3)$$

A výsledné pásmo, jak už bylo zmíněno, se spočítá jako dvojnásobek této vzdálenosti. Příklad takového pásma je znázorněn v obrázku 4.2.[9]

$$margin = \frac{2}{\|\beta\|} \quad (4.4)$$



Obrázek 4.2: Ukázka pásma a rozdělovací nadroviny pro dvě třídy

Následně se musí zjistit velikost maximálního pásma. Tento optimalizační problém je potřeba převést do tvaru, který je možné efektivně vyřešit. K tomu je využito faktu, že maximalizace $\frac{1}{\|\beta\|}$ je stejná jako minimalizace $\|\beta\|^2$. Využitím tohoto faktu dostaneme optimalizační problém

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 \text{ subject to } y_i(\beta^T x_i + \beta_0) \geq 1 \forall i, \quad (4.5)$$

kde štítky jednotlivých tříd trénovacích dat jsou označeny jako y_i . Tento optimalizační problém je kvadratickým optimalizačním problémem s lineárními omezeními, který může být vyřešen pomocí Lagrangian násobitelů (Lagrangian multipliers).[10]

Nelineárně rozdělitelná data

Při hledání optimální rozdělovací nadroviny pro jednotlivé třídy může nastat situace, kdy nejsou data lineárně rozdělitelná a nadrovina bude některá data klasifikovat špatně. Další možnou situací jsou lineárně rozdělitelná data, pro která když najdeme optimální rozdělení, vznikne příliš malé pásmo náchylné na šum trénovacích dat blízkých k nadrovině. Proto je nutné tuto špatnou klasifikaci brát v úvahu při optimalizaci. Cílem nelineární klasifikace by mělo být nalezení rozdělovací nadroviny tak aby měla co největší pásmo a zároveň aby byla data generalizována správně a počet chyb v klasifikaci byl co nejmenší.

K začlenění chyb v klasifikaci do dříve uvedeného optimalizačního problému se využije modifikace pomocí konstanty C krát vzdálenost chybně klasifikovaných dat od oblastí jejich správného zařazení.

$$\min \|\beta\|^2 + C(\text{distance}) \quad (4.6)$$

Tato vzdálenost je označena jako parametr ξ_i a je definována pro každý vzorek trénovacích dat. U všech dobře klasifikovaných vzorků je tato vzdálenost rovna nule.

Využitím výše popsaného je dosažen upravený optimalizační problém

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_i \xi_i \text{ subject to } y_i(\beta^T x_i + \beta_0) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \forall i. \quad (4.7)$$

Parametr C je regulační parametr. Umožňuje regulovat velikost pásma a množství klasifikačních chyb, avšak tyto dvě veličiny nelze regulovat odděleně. Pokud je pro C zvolena vysoká hodnota, bude výsledkem menší počet klasifikačních chyb, ale pásmo, které je získáno, bude menší. A naopak pro menší hodnoty parametru C je výsledkem větší počet klasifikačních chyb, ale pásmo je větší. Další možností je zvolit hodnotu parametru C nekonečno čímž je získáno tvrdé pásmo, které nedovoluje žádné chyby.[11]

4.1.2 Kernel

Optimalizační problém vyřešený pomocí Langrangian násobitelů lze napsat celý s využitím skalárního součinu $\langle x_i, x_j \rangle$, který může být brán jako $(x_i)^T (x_j)$. Tento skalární součin může být nahrazen skalárním součinem $\langle \phi(x_i), \phi(x_j) \rangle$, kde ϕ označuje mapování příznaků. Odpovídající jádrová funkce je dána vztahem

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j). \quad (4.8)$$

Poté je nahrazen skalární součin $\langle x_i, x_j \rangle$ pomocí $K(x_i, x_j)$ a výsledkem je algoritmus fungující na základě mapování příznaků.

Například lineární jádrová funkce je dána vztahem

$$K(x_i, x_j) = x_i^T x_j. \quad (4.9)$$

Velkou výhodou těchto jádrových funkcí je fakt, že existuje efektivní způsob jak je vypočítat bez nutnosti explicitně hledat nebo reprezentovat vektory $\phi(x)$, které můžou být velmi náročné na výpočet.[10]

4.1.3 Nelineární SVM

Jedním ze základních principů je převod vstupního prostoru, v kterém nelze data rozdělit lineárně, do prostoru, který má více dimenzí a lze v tomto prostoru data rozdělit lineárně. Tento převod se provádí pomocí mapování příznaků. Tak jako existuje dříve uvedená jádrová funkce pro lineární rozdělení dat v původním prostoru, tak existují i jádrové funkce pro nelineární rozdělení dat pomocí mapování příznaků.[12]

Polynomiální jádrová funkce

$$K(x_i, x_j) = (\gamma x_i^T x_j + coef0)^{degree}, \gamma > 0. \quad (4.10)$$

Radial basis funkce (RBF)

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \gamma > 0. \quad (4.11)$$

Jádrová funkce Sigmoid

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + coef0), \gamma > 0. \quad (4.12)$$

4.2 Další klasifikátory

Klasifikátor SVM byl popsán podrobněji z důvodu jeho následného použití při implementaci algoritmu pro detekci obsazenosti parkovacích míst. Avšak pro klasifikaci existují i jiné klasifikátory jako například:

- Normal Bayes klasifikátor – používá ke klasifikaci vektory průměrů a kovarianční matice, které odhadne z trénovacích dat.¹

¹ K. Fukunaga. *Introduction to Statistical Pattern Recognition*. second ed., New York: Academic Press, 1990.

- K-Nearest Neighbors – provádí klasifikaci na základě analýzy daného počtu (K) nejbližších sousedů.
- Decision Trees – binární stromy, kde je pro klasifikaci každý list stromu označen podle třídy, do které patří.²
- Random Trees – skupina rozhodovacích stromů, které dohromady tvoří les. Klasifikace je poté prováděna jako průměr odpovědí všech stromů v lese.³
- Boosting.⁴
- Neural Networks.⁵

² Breiman, L., Friedman, J. Olshen, R. and Stone, C. (1984), *Classification and Regression Trees*, Wadsworth

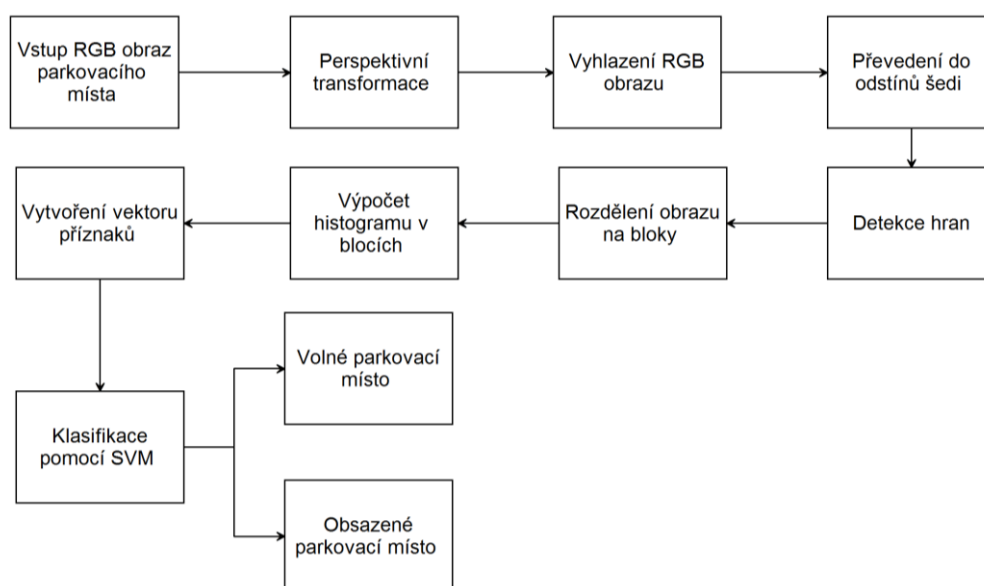
³ <http://www.stat.berkeley.edu/users/breiman/RandomForests/>

⁴ Hastie, T., Tibshirani, R., Friedman, J. H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. 2001.

⁵ Y. LeCun, L. Bottou, G.B. Orr and K.-R. Muller, *Efficient backprop*, in Neural Networks—Tricks of the Trade, Springer Lecture Notes in Computer Sciences 1524, pp.5-50, 1998.

5 Popis použitého algoritmu

Použitý algoritmus je inspirován algoritmem Histogram orientovaných gradientů, který použili Navneet Dalal a Bill Triggs ve své práci [5]. Hlavním rozdílem mezi Histogramem orientovaných gradientů a algoritmem použitým v této práci je, že se místo orientace gradientu nebo orientace hrany využívá pouze velikosti gradientu a dále se nepoužívá normalizace buněk pomocí větších bloků, ale jsou použity pouze samotné bloky.



Obrázek 5.1: Grafické znázornění použitého algoritmu

Vstupem pro algoritmus je RGB obraz celého parkoviště, které je snímáno kamerou umístěnou na střeše budovy v areálu Vysoké školy báňské. Umístění kamery je možné vidět na obrázku 5.2.



Obrázek 5.2: Umístění kamery snímající parkoviště

Poté se pomocí dané geometrie obraz parkoviště rozdělí na jednotlivá parkovací místa, čímž se vytvoří oblasti zájmu (Region Of Interest ROI), aby nebylo potřeba procházet celý obraz a hledat jednotlivá parkovací místa. Avšak je potřebné mít určenou geometrii pro dané parkoviště.



Obrázek 5.3: Ukázka rozdělení parkoviště na jednotlivá parkovací místa

Na takto získaná parkovací místa je použita perspektivní transformace. Která je provedena jako

$$dst(x, y) = src\left(\frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}}, \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}}\right), \quad (5.1)$$

kde M je specifikovaná 3x3 matice pro perspektivní transformaci.

Dále se v algoritmu pracuje pouze s jednotlivými parkovacími místy. Transformovaný RGB obraz parkovacího místa je následně vyhlazen pomocí aplikování lineárního filtru. Pro testování byly použity dva typy filtrů:

- Medián filtr, který počítá medián sousedících pixelů a následně touto hodnotou nahradí původní hodnotu pixelu, kolem kterého byl medián počítán a tento postup aplikuje na všechny pixely v obraze.
- Gaussián filtr provádí konvoluci mezi konvoluční maskou pro Gaussián a obrazem.

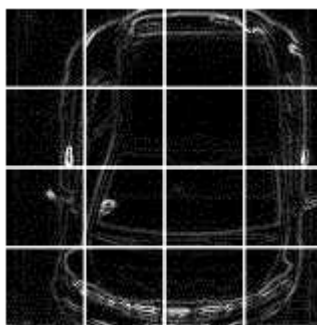
Rozdíl mezi filtry je ukázán na obrázku 5.4.



Obrázek 5.4: Rozdíl mezi vyhlazením obrazu mediánem (vlevo) a Gaussiánem (vpravo)

Obraz získaný po vyhlazení se převede z RGB do odstínu šedi. Ukázkou takto převedeného parkovacího místa je obrázek 1.1. Následuje výpočet detekce hran v obraze. Hrany jsou detekovány pomocí dvou různých operátorů. Prvním z těchto operátorů je Sobelův operátor a tím druhým je Laplaceův operátor. Oba tyto operátory byly popsány v kapitole 1 a na obrázcích 1.3, 1.4 lze vidět rozdíl mezi výsledkem po použití obou operátorů na stejný obraz.

Po detekování hran se výsledný obraz obsahující hrany rozdělí na menší části obrazu, které se nazývají bloky. Pro testovací účely byly použity různé velikosti bloků a to konkrétně 4x4, 8x8, 16x16 a 32x32 pixelů. Příkladem rozdělení obrazu na bloky je obrázek 5.5. V obrázku je použito rozdělení pro přehlednost pouze na bloky o velikost 40x40 pixelů.



Obrázek 5.5: Rozdělení obrazu na bloky

V takto rozděleném obraze následuje výpočet histogramu gradientu ve všech blocích obrazu zvlášť. Zkombinováním histogramů gradientů ze všech bloků vznikne vektor příznaků, který obsahuje za sebou jdoucí normalizované histogramy gradientů bloků. Histogramy jsou normalizovány počtem pixelů, které náleží jednotlivým blokům. Výsledný deskriptor je potom dán tímto vektorem příznaků, který slouží jako vstup do klasifikátoru.

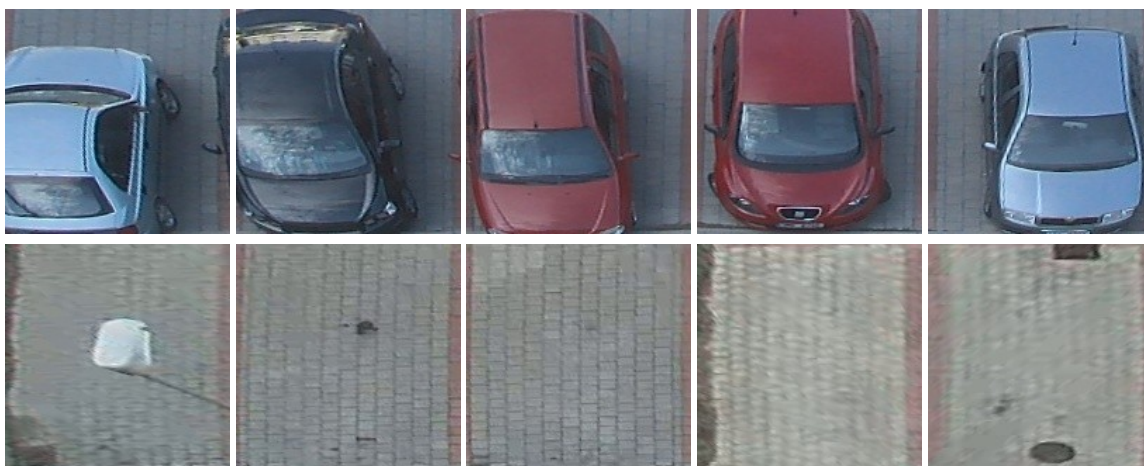
Jako klasifikátor byl použit lineární Support Vector Machine (SVM). Jehož výstupem je označení místa buď za obsazené, nebo volné. Výstup detektoru je ukázán na obrázku 5.6.



Obrázek 5.6: Výsledek výstupu detektoru

5.1 Proces trénování

K trénování a následnému testování detektoru byl použit soubor dat o celkovém počtu 121 snímků parkoviště. Jedno parkoviště obsahuje 56 míst k parkování. Takže k samotnému trénování a testování bylo využito 6776 parkovacích míst. Z toho 1120 testovacích a 5656 trénovacích parkovacích míst. Trénovací set byl poskládán z 2688 prázdných míst pro negativní trénovací případy a z 2968 obsazených míst pro pozitivní případy.



Obrázek 5.7: Pozitivní (nahore) a negativní (dole) trénovací data

6 Experimenty

Všechny experimenty byly prováděny nad testovacím setem, který obsahoval 20 snímků parkoviště, to odpovídá 1120 testovaným parkovacím místům. Z těchto testovaných parkovacích míst bylo 594 pozitivních snímků, neboli obsazených míst a 526 negativních snímků, které odpovídaly prázdným parkovacím místům. Do experimentů byly zařazeny dvě metody pro detekci hran v obraze a dvě metody pro vyhlazení obrazu a následně byla testována přesnost různých kombinací detekcí hran a vyhlazení obrazu. Dále byl proveden experiment s různými velikostmi bloků rozdělujících obraz. A nakonec byl proveden pokus se změnou parametru C lineárního SVM.

Pro určení efektivity jednotlivých detektorů byl použit výpočet přesnosti podle vzorce

$$přesnost = \frac{\text{Správně pozitivní} + \text{Správně negativní}}{\text{Správně pozitivní} + \text{Správně negativní} + \text{Chybně pozitivní} + \text{Chybně negativní}}. \quad (6.1)$$

Histogram v jednotlivých blocích obrazu byl vypočten pro rozdělení na 10 košů. Dimenzionalita vektoru příznaků uvedená v tabulkách 6.1, 6.3, 6.5, 6.7 byla následně vypočtena vztahem

$$dimenzionalita = \left(\frac{\text{velikost okna}}{\text{velikost bloku}} \right)^2 \times \text{počet košů}. \quad (6.2)$$

Jako výchozí detektor byl použit detektor využívající pro detekci hran Sobelův operátor popsáný v kapitole 1.3, bez vyhlazení obrazu a s velikostí bloků 8x8 pixelů. Velikost okna pro detekci byla zvolena s ohledem na možnost vytvoření velikostí bloků o velikostech násobků dvou až po velikost 32x32 pixelů na 160x160 pixelů. Klasifikaci výchozího detektoru zajišťoval jemný lineární SVM s parametrem C nastaveným na hodnotu 0,1. K čemu slouží parametr C je popsáno v kapitole 4.1 o SVM. Přesnost tohoto výchozího detektoru byla 95,9%. K dalšímu testování byly použity různé modifikace tohoto detektoru, které jsou popsány v následujících kapitolách.

6.1 Použité operátory

Prvním operátorem použitým pro detekci hran je Sobelův operátor, který je využit ve výchozím detektoru a druhým operátorem je Laplaceův operátor. Oba operátory a metody, které využívají, jsou popsány v kapitole 1. Při testování byl ve výchozím detektoru Sobelův operátor nahrazen Laplaceovým. Nahrazením došlo k snížení přesnosti detektoru o 0,9%. To bylo způsobeno šumem, který vzniká při použití Laplaceova operátoru, jak už bylo zmíněno v kapitole 1.2.

Tabulka 6.1: Experimenty s operátory pro detekci hran

Experiment	Detektor	Dimenzionalita vektoru příznaků
1	8x8 Sobel $C=0,1$	4000
2	8x8 Laplacian $C=0,1$	4000

Tabulka 6.2: Výsledky experimentů s operátory pro detekci hran

Experiment	Správně pozitivní	Správně negativní	Chybně pozitivní	Chybně negativní	Přesnost (%)	Čas detekce (s)
1	590	484	42	4	95,9	0,251
2	574	490	36	20	95	0,257

6.2 Kombinace detekcí hran a vyhlazení

K vyhlazení obrazu byly použity dva druhy filtrování a to konkrétně filtrování Gaussiánem a mediánem. Oba způsoby vyhlazení byly zkombinovány s operátory, pro které byly provedeny experimenty v kapitole 6.1.

Tabulka 6.3: Experimenty s kombinacemi detekce hran a vyhlazení

Experiment	Detektor	Dimenzionalita vektoru příznaků
3	8x8 Gaussian Sobel $C=0,1$	4000
4	8x8 Median Sobel $C=0,1$	4000
5	8x8 Gaussian Laplacian $C=0,1$	4000
6	8x8 Median Laplacian $C=0,1$	4000

Tabulka 6.4: Výsledky experimentů s kombinacemi detekce hran a vyhlazení

Experiment	Správně pozitivní	Správně negativní	Chybně pozitivní	Chybně negativní	Přesnost (%)	Čas detekce (s)
3	527	503	23	67	91,96	0,283
4	554	501	25	40	94,2	0,583
5	589	481	45	5	95,54	0,27
6	592	507	19	2	98,13	0,57

6.2.1 Vyhlazení před použitím Sobelova operátoru

Při použití filtrace Gaussiánem přidané do výchozího detektoru před samotnou detekcí hran se přesnost detektoru oproti výchozímu detektoru značně snížila a to o 3,94%. Vyhlazení mediánem bylo v přesnosti detekce účinnější než vyhlazení Gaussiánem, ale v porovnání s výchozím detektorem měl tento detektor o 1,7% horší přesnost. Použití těchto dvou metod pro vyhlazení obrazu vedlo v kombinaci se Sobelovým operátorem ke zhoršení přesnosti detektorů.

6.2.2 Vyhlazení před použitím Laplaceova operátoru

Detektor využívající Laplaceova operátoru bez vyhlazení před jeho aplikací na obraz dosahoval přesnosti 95%. Po přidání vyhlazení Gaussiánem před použití operátoru se přesnost mírně zvýšila o 0,54%, ale stále měl tento detektor menší přesnost než detektor využívající pouze Sobelův operátor. Dalším experimentem bylo zkombinování vyhlazení mediánem a Laplaceovým operátorem. U této kombinace byl zaznamenán značný nárůst přesnosti o 2,23% oproti dosud nejefektivnějšímu detektoru využívajícího Sobelova operátoru bez vyhlazení obrazu. Tato kombinace dosáhla přesnosti 98,13%. V porovnání se Sobelovým operátorem, kde vyhlazení obrazu v obou případech vedlo k snížení přesnosti, tak v případě Laplaceova operátoru se stal přesný opak a to že využití vyhlazení obrazu vedlo k zlepšení přesnosti detektoru.

Z testovaných kombinací operátorů pro detekci hran a způsobů vyhlazení obrazu je možné posoudit výkonnost dvou využitých vyhlazení obrazu. V případě vyhlazení Gaussiánem v kombinaci se Sobelovým operátorem bylo snížení přesnosti větší než v případě vyhlazení mediánem. Jelikož oproti využití vyhlazení obrazu se Sobelovým operátorem se v případě použití Laplaceova operátoru přesnost zvyšovala, je brán jako méně efektivní způsob ten, který vedl k menšímu nárůstu přesnosti. Vyhlazení mediánem vedlo k většímu zvýšení přesnosti než vyhlazení Gaussiánem. A protože v obou situacích si vedlo lépe vyhlazení mediánem, je možné tvrdit, že pro testovanou problematiku je vyhlazení mediánem efektivnější než vyhlazení Gaussiánem.

6.3 Velikost bloků

Pro experimenty vlivu velikosti bloků na efektivitu detektorů byly vybrány dva detektory, které v předchozích experimentech dosahovaly nejvyšších přesností. Jedním z nich byl výchozí detektor s následujícími parametry:

- velikost bloků 8x8 pixelů,
- detekce hran pomocí Sobelova operátoru,
- bez vyhlazení obrazu,
- klasifikací prováděnou jemným lineárním SVM ($C=0,1$).

Druhý detektor, který dosáhl také nejlepší přesnosti, s těmito parametry:

- velikost bloků 8x8 pixelů,
- detekce hran pomocí Laplaceova operátoru,
- s vyhlazením obrazu mediánem,
- klasifikací prováděnou jemným lineárním SVM ($C=0,1$).

Experimenty byly provedeny pro další dvě velikosti bloků a to pro velikost bloků 16x16 pixelů a velikost 32x32 pixelů. Změny ve velikostech bloků vedou k velkým změnám v počtu bloků sloužících k výpočtu histogramů gradientu, a tím se snižuje přesnost popisu vzhledu a tvaru objektů. Například při změně z velikosti bloků 8x8 pixelů na velikost 32x32 pixelů se počet bloků 16krát zmenší. A to ze 400 bloků na 25 bloků.

Tabulka 6.5: Experimenty s velikostmi bloků

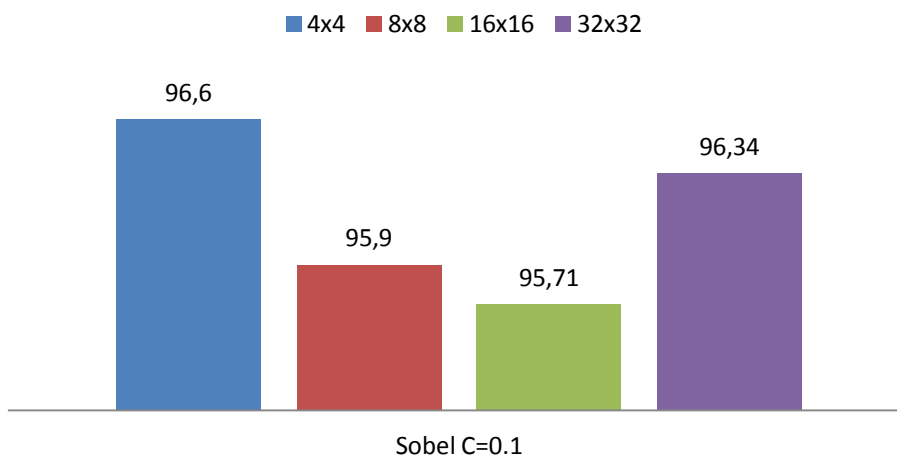
Experiment	Detektor	Dimenzionalita vektoru příznaků
7	16x16 Median Laplacian $C=0,1$	1000
8	32x32 Median Laplacian $C=0,1$	250
9	4x4 Median Laplacian $C=0,1$	16000
10	16x16 Sobel $C=0,1$	1000
11	32x32 Sobel $C=0,1$	250
12	4x4 Sobel $C=0,1$	16000

Tabulka 6.6: Výsledky experimentů s velikostmi bloků

Experiment	Správně pozitivní	Správně negativní	Chybně pozitivní	Chybně negativní	Přesnost (%)	Čas detekce (s)
7	592	504	22	2	97,86	0,465
8	589	509	17	5	98,04	0,44
9	591	510	16	3	98,3	0,97
10	589	483	43	5	95,71	0,15
11	586	493	33	8	96,34	0,123
12	590	492	34	4	96,6	0,65

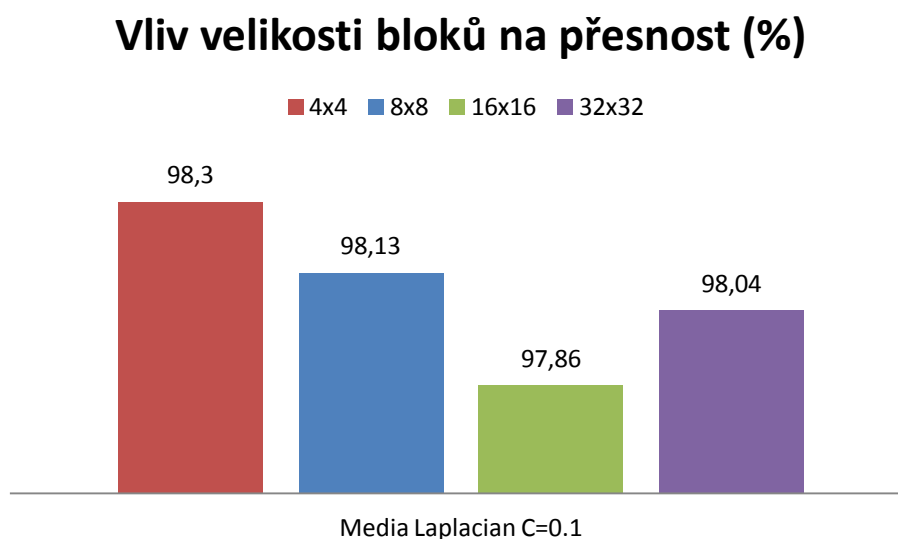
U prvního detektoru byla při použití velikosti bloků 8x8 pixelů přesnost 95,9%. Po zvýšení velikosti bloků na 16x16 se přesnost detektoru mírně snížila a to o 0,12%. Následně byl proveden experiment se zvýšením velikosti na 32x32 pixelů. Toto navýšení velikosti nečekaně vedlo ke zvýšení přesnosti, a to tak, že tento detektor dosáhl přesnosti dokonce vyšší než při použití velikosti 8x8 pixelů. Přesnost tohoto detektoru byla 96,34%.

Vliv velikosti bloků na přesnost (%)



Obrázek 6.1: Graf vlivu velikosti bloků na přesnost detektoru využívajícího Sobelův operátor

U druhého detektoru bylo dosaženo jiných výsledků v porovnání s prvním. Druhý detektor s velikostí bloků 8x8 pixelů měl přesnost 98,13% a zvyšování velikosti bloků na 16x16 a 32x32 vedlo k snížení přesnosti v obou případech. V případě velikosti bloků 16x16 pixelů došlo ke snížení přesnosti o 0,27%. Při použití velikosti 32x32 pixelů došlo stejně jako u prvního detektoru k zvýšení přesnosti oproti blokům 16x16 pixelů, ale v porovnání s bloky 8x8 pixelů byla přesnost detektoru nižší. Rozdíl v přesnosti mezi těmito dvěma velikostmi bloků byl 0,09%.



Obrázek 6.2: Graf vlivu velikosti bloků na přesnost detektoru využívajícího Laplaceův operátor

Následně byl proveden experiment s bloky o velikosti 4x4. Toto další snížení velikosti bloků vedlo k zvýšení přesnosti u druhého detektoru o 0,17%, ale za cenu vyššího času detekce z důvodu navýšení počtu bloků 4krát a to ze 400 na 1600 bloků. A u prvního detektoru tato velikost bloků také vedla k zvýšení přesnosti oproti všem doposud použitým velikostem a to o 0,26%, avšak také došlo k navýšení času detekce.

V obou detektorech měla velikost bloků 16x16 nejhorší vliv na výslednou efektivitu. Ale podle výsledků ze zbylých dvou bylo zjištěno, že pro druhý detektor jsou nižší velikosti bloků efektivnější až na 16x16 pixelů. Avšak první detektor dosahoval vyšších přesností při velikosti bloků 32x32 pixelů než při 8x8 pixelů. Vliv na přesnost velikostí bloků 8x8 a 32x32 závisí na použitých parametrech detektoru.

Nejlepší vliv na výslednou přesnost detekce měla velikost bloků 4x4, ale jak už bylo řečeno při použití těchto bloků došlo v obou případech k navýšení času detekce.

6.4 Parametr C u lineárního SVM

Pro experimenty s parametrem C u lineárního SVM byly vybrány stejné detektory jako v případě experimentů s velikostmi bloků. Pro všechny předchozí experimenty byl použit parametr C o hodnotě 0,1. Podle nastavení parametru C na 0,01, které použili pro jemný lineární SVM Navneet Dalal a Bill Triggs ve své práci [5], bylo stejné nastavení využito pro experiment.

Tabulka 6.7: Experimenty s parametrem C lineárního SVM

Experiment	Detektor	Dimenzionalita vektoru příznaků
13	8x8 Median Laplacian C=0,01	4000
14	8x8 Sobel C=0,01	4000
15	4x4 Median Laplacian C=0,01	16000
16	8x8 Median Laplacian(C=1)	4000
17	8x8 Sobel (C=1)	4000

Tabulka 6.8: Výsledky experimentů s parametrem C lineárního SVM

Experiment	Správně pozitivní	Správně negativní	Chybně pozitivní	Chybně negativní	Přesnost (%)	Čas detekce (s)
13	588	516	10	6	98,57	0,575
14	586	491	35	8	96,16	0,263
15	590	512	14	4	98,39	0,987
16	591	507	19	3	98,04	0,572
17	591	484	42	3	95,98	0,264

U obou detektorů bylo zaznamenáno navýšení přesnosti. První detektor měl původně přesnost 95,9% a po změně parametru C se jeho přesnost zvýšila na 96,16%. A druhý detektor se z přesnosti 98,13% dostal na přesnost 98,57%. Následoval experiment s navýšením parametru C a to na hodnotu 1.

V tomto experimentu se přesnost u prvního detektoru zvýšila o 0,08% a u druhého se snížila o 0,09%. I přes navýšení efektivity prvního detektoru po zvýšení parametru C byla jeho přesnost stále nižší než pro parametr C o hodnotě 0,01. Proto pro oba detektory bylo snížení hodnoty parametru C efektivnější než jeho zvýšení.

Následoval experiment s detektorem, který v předchozích experimentech dosáhl při hodnotě parametru $C=0,1$ nejlepší přesnosti. Jedná se o experiment číslo 9 v tabulce 6.6. Pro tento detektor byla snížena hodnota parametru C na $0,01$. Výsledkem bylo navýšení přesnosti na $98,39\%$.

6.5 Chybně pozitivní a negativní

Ze všech detektorů použitých pro experimenty měla většina větší počet chybně pozitivních detekcí než chybně negativních, a to ve všech případech o nezanedbatelné množství. Jediné dva detektory, které měly větší počet chybně negativních než chybně pozitivních, byly detektory z kapitoly 6.1 pro experimenty vyhlazení obrazu před použitím Sobelova operátoru. Na obrázku 6.3 je možné vidět ukázkou chybné detekce a to jak chybně negativní tak i chybně pozitivní.



Obrázek 6.3: Ukázkou chybně pozitivní a chybně negativní detekce

6.6 Čas detekce parkoviště

Pro měření času detekce byl použit notebook Lenovo ThinkPad SL510 s procesorem Intel Mobile Core 2 Duo T6670 2.20GHz a 32 bitovým operačním systémem Windows 7. Čas detekce celého parkoviště u žádného detektoru nepřekročil dobu jedné vteřiny. Nejrychlejší byla detekce prováděná detektorem 32×32 Sobel $C=0,1$, která trvala $0,123$ vteřin. Naopak nejpomalejší byl detektor 4×4 Median Laplacian $C=0,01$ jehož detekce trvala $0,987$ vteřin. Čas detekce všech detektorů je zaznamenán v tabulkách 6.2, 6.4, 6.6, 6.8.

Závěr

V práci bylo dosaženo stanoveného cíle a to vytvoření detektoru obsazenosti parkovacích míst. Byl otestován vliv změn částí algoritmu na přesnost detektoru. Z testovaných kombinací vyhlazení obrazu pro redukci šumu a detekci hran měla nejlepší výsledky kombinace Laplaceova operátoru a vyhlazení mediánem. Nejefektivnější velikostí bloků byla pro tuto kombinaci velikost 4x4 pixely. Avšak po experimentech s parametrem C u lineárního klasifikátoru Support Vector Machine se ukázala jako přesnější velikost bloků 8x8 pixelů v kombinaci s hodnotou 0,01 parametru C . Detektor s tímto nastavením dosáhl nejlepší přesnosti 98.57%. Čas detekce detektoru byl naměřen 0,575 vteřin. Dosaženými výsledky byla otestována úspěšnost použitého přístupu. Největší počet chybně pozitivních detekcí byl zaznamenán u míst, kde do prostoru parkovacího místa zasahovalo z pohledu kamery auto parkující na vedlejším parkovacím místě. Naopak chybně negativních detekcí bylo nejvíce zaznamenáno v řadě nejdále od kamery. V těchto místech byla chybně detekována místa jako prázdná v případech, kdy parkující auto zabíralo z pohledu kamery pouze malou část parkovacího místa. Snížení počtu těchto chybných detekcí by vedlo k optimalizaci detektoru. Jedním z možných vylepšení detektoru by mohlo být přidání informace o jasu do vektoru příznaků. Další případné zdokonalení by mohlo nastat při přetrénování klasifikátoru za pomoci trénovacího setu, který by obsahoval některé chybně negativní a chybně pozitivní detekce parkovacích míst.

Použitá literatura

- [1] SOJKA, Eduard, GAURA, Jan, KRUMNIKL Michal. *Matematické základy digitálního zpracování obrazu* [online]. Ostrava: Vysoká škola báňská – Technická univerzita, 2011 [cit. 2014-03-06]. Matematika pro inženýry 21. století. Dostupné z: http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/digitalni_zpracovani_obrazu.pdf.
- [2] HUAMÁN, Ana. Sobel Derivates. In: *OpenCV Tutorials* [online]. OpenCV.org ©2011-2014 [cit. 2014-03-08]. OpenCV Documentation. Dostupné z: http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/sobel_derivatives/sobel_derivatives.html.
- [3] BRADSKI, Gary, KAEHLER, Adrian. *Learning OpenCV* [online]. First Edition. Sebastopol (CA, USA): O'Reilly Media, 2008 [cit. 2014-03-11]. Dostupné také z: <http://it-ebooks.info/book/299/>. ISBN 978-0-596-51613-0.
- [4] HUAMÁN, Ana. Histogram Calculation. In: *OpenCV Tutorials* [online]. OpenCV org. ©2011-2014 [cit. 2014-03-14]. OpenCV Documentation. Dostupné z: http://docs.opencv.org/doc/tutorials/imgproc/histograms/histogram_calculation/histogram_calculation.html.
- [5] DALAL, Navneet, TRIGGS, Bill. Histograms of oriented gradients for human detection. In: *Computer Vision and Pattern Recognition 2005* [online]. CVPR, San Diego (CA, USA), 2005 [cit. 2014-03-18]. IEEE Computer Society Conference on, p. 886-893 vol. 1 (25 June 2005). Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1467360>. ISSN 1063-6919.
- [6] OpenCV. Introduction to SIFT (Scale-Invariant Feature Transform). In: *OpenCV-Python Tutorials* [online]. OpenCV.org ©2011-2014 [cit. 2014-03-21]. OpenCV Documentation. Dostupné z: http://docs.opencv.org/trunk/doc/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html.
- [7] LOWE, David. Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision*. Kluwer Academic Publishers (Netherlands), 2004 [2014-03-23]. Volume 60, Issue 2, p. 91-110, november 2014. Dostupné také z: http://download.springer.com/static/pdf/941/art%253A10.1023%252FB%253AVISI.0000029664.99615.94.pdf?auth66=1398806195_61c9c715319c9252e2fc4b20779c1512&ext=.pdf. ISSN 1573-1405.
- [8] BREHAR, Raluca. Histogram of Oriented Gradients. In: *Pattern recognition systems* [online]. Technical University Of Cluj-Napoca (Romania), 29. Oct. 2012 [cit. 2014-03-26]. Lecture notes. Dostupné z: http://users.utcluj.ro/~igiosan/Resources/PRS/L5/lab_05e.pdf.

- [9] GARCÍA, Fernando. Introduction to Support Vector Machines. In: *OpenCV Tutorials* [online]. ©2011-2014 [cit. 2014-03-29]. OpenCV Documentation. Dostupné z: http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html.
- [10] NG, Andrew. Support Vector Machines. In: *Machine Learning* [online]. Stanford University, Stanford (CA, USA) [cit. 2014-04-03]. Lecture notes. Dostupné z: <http://cs229.stanford.edu/notes/cs229-notes3.pdf>.
- [11] GARCÍA, Fernando. Support Vector Machines for Non-Linearly Separable Data. In: *OpenCV Tutorials* [online]. OpenCV.org ©2011-2014 [cit. 2014-04-05]. OpenCV Documentation. Dostupné z: http://docs.opencv.org/doc/tutorials/ml/non_linear_svms/non_linear_svms.html.
- [12] OpenCV. Support Vector Machines. In: *OpenCV API Reference* [online]. OpenCV.org ©2011-2014 [cit. 2014-04-09]. OpenCV Documentation. Dostupné z: http://docs.opencv.org/modules/ml/doc/support_vector_machines.html.

Seznam obrázků

Obrázek 1.1: Převedení originálního obrazu (vlevo) do odstínů šedi (vpravo)	10
Obrázek 1.2: Výsledek konvolučních masek pro horizontální (vlevo) a vertikální (vpravo) směr ..	11
Obrázek 1.3: Výsledek po vypočtení velikosti gradientu.....	11
Obrázek 1.4: Výsledek detekce hran Laplaceovým operátorem.....	13
Obrázek 3.1: Ukázka problému nastávajícího při detekci rohů	15
Obrázek 3.2: Grafické znázornění HOG algoritmu	16
Obrázek 4.1: Několik možností pro rozdělení dvou tříd přímkou	17
Obrázek 4.2: Ukázka pásma a rozdělovací nadroviny pro dvě třídy	18
Obrázek 5.1: Grafické znázornění použitého algoritmu	22
Obrázek 5.2: Umístění kamery snímající parkoviště	23
Obrázek 5.3: Ukázka rozdělení parkoviště na jednotlivá parkovací místa.....	23
Obrázek 5.4: Rozdíl mezi vyhlazením obrazu mediánem (vlevo) a Gaussiánem (vpravo)	24
Obrázek 5.5: Rozdělení obrazu na bloky	25
Obrázek 5.6: Výsledek výstupu detektoru.....	25
Obrázek 5.7: Pozitivní (nahore) a negativní (dole) trénovací data	26
Obrázek 6.1: Graf vlivu velikosti bloků na přesnost detektoru využívajícího Sobelův operátor	31
Obrázek 6.2: Graf vlivu velikosti bloků na přesnost detektoru využívajícího Laplaceův operátor ..	32
Obrázek 6.3: Ukázka chybně pozitivní a chybně negativní detekce	34

Seznam tabulek

Tabulka 1.1: Označení sousedících pixelů pro operátor Prewittové	9
Tabulka 1.2: Konvoluční maska Laplaceova operátora	12
Tabulka 1.3: Konvoluční maska pro výpočet hodnot v dolním řádku obrazu	12
Tabulka 1.4: Konvoluční maska pro výpočet v levém horním rohu obrazu	13
Tabulka 6.1: Experimenty s operátory pro detekci hran	28
Tabulka 6.2: Výsledky experimentů s operátory pro detekci hran	28
Tabulka 6.3: Experimenty s kombinacemi detekce hran a vyhlazení	28
Tabulka 6.4: Výsledky experimentů s kombinacemi detekce hran a vyhlazení	29
Tabulka 6.5: Experimenty s velikostmi bloků	30
Tabulka 6.6: Výsledky experimentů s velikostmi bloků	31
Tabulka 6.7: Experimenty s parametrem C lineárního SVM	33
Tabulka 6.8: Výsledky experimentů s parametrem C lineárního SVM	33

Seznam příloh

Příloha 1.: Detektor obsazenosti parkovacích míst (Příloha na DVD)

Obsah DVD: Detektor, Geometry, Pictures, SVM, Training Data Matrix.